**Entities:**

1. User

- Attributes:

- User\_ID (Primary Key)

- Username

- Email

- Password

- First Name

- Last Name

- Profile Picture

- Registration Date

2. Blog

- Attributes:

- Blog\_ID (Primary Key)

- Title

- Content

- Creation Date

- Last Modified Date

- Published (Boolean)

- User\_ID (Foreign Key referencing User)

3. Category

- Attributes:

- Category\_ID (Primary Key)

- Name

4. Tag

- Attributes:

- Tag\_ID (Primary Key)

- Name

5. Comment

- Attributes:

- Comment\_ID (Primary Key)

- Content

- Creation Date

- User\_ID (Foreign Key referencing User)

- Blog\_ID (Foreign Key referencing Blog)

6. Like

- Attributes:

- Like\_ID (Primary Key)

- User\_ID (Foreign Key referencing User)

- Blog\_ID (Foreign Key referencing Blog)

7. Follow

- Attributes:

- Follow\_ID (Primary Key)

- Follower\_ID (Foreign Key referencing User)

- Following\_ID (Foreign Key referencing User)

8. Report

- Attributes:

- Report\_ID (Primary Key)

- Content

- Creation Date

- User\_ID (Foreign Key referencing User)

- Blog\_ID (Foreign Key referencing Blog)

Relationships:

- Each User can write multiple Blogs (One-to-Many from User to Blog)

- Each Blog belongs to one User (Many-to-One from Blog to User)

- Each Blog can have multiple Categories (Many-to-Many between Blog and Category, implemented via an intermediate table)

- Each Blog can have multiple Tags (Many-to-Many between Blog and Tag, implemented via an intermediate table)

- Each Blog can have multiple Comments (One-to-Many from Blog to Comment)

- Each Comment is written by one User (Many-to-One from Comment to User)

- Each Comment belongs to one Blog (Many-to-One from Comment to Blog)

- Each Blog can receive multiple Likes (One-to-Many from Blog to Like)

- Each Like is given by one User (Many-to-One from Like to User)

- Each User can give multiple Likes (One-to-Many from User to Like)

- Each User can follow multiple other Users (Many-to-Many between User and User, implemented via an intermediate table)

- Each User can report multiple Blogs (One-to-Many from User to Report)

- Each Report is associated with one User (Many-to-One from Report to User)

- Each Report is associated with one Blog (Many-to-One from Report to Blog)

Attributes for Normalization:

1. Ensure that each attribute is atomic (indivisible).

2. Minimize repeating groups by separating them into distinct entities.

3. Identify and remove any redundant data.

Normalization helps to eliminate data redundancy and maintain data integrity. The database design should aim for at least 3rd normal form (3NF) to ensure efficient storage and retrieval of information.